THE REMARKS

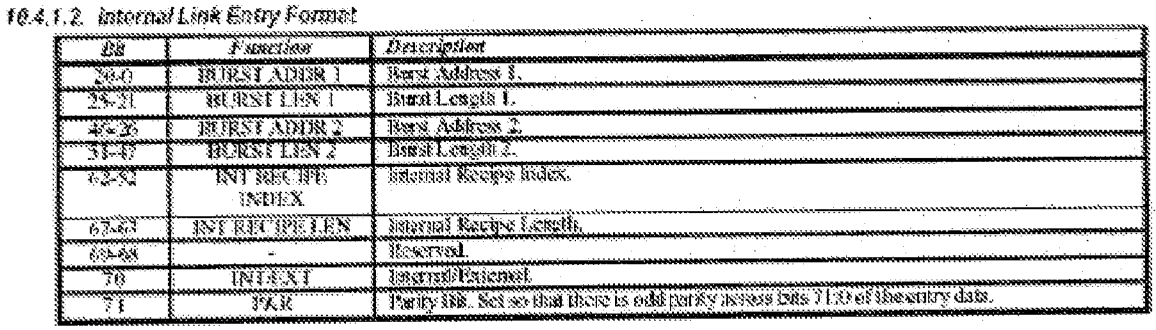
Claims 1-3, 6-15, 18-22, 24 and 28-30 are pending in the application. Claims 4-5, 16-17, 23 and 25-27 are cancelled. Claims 1, 6, 9, 11, 18, 21, 22, 24 and 30 are amended herein.

The Amendments

Claim 1, directed to a data structure, has been amended to clarify that both the first pointer (pointing to a sequence of commands stored in a first memory area) and the second pointer (pointing to a burst of data or mask items stored in a second memory area distinct from the first) are entered into the same data structure. These amendments are set forth in the first and second paragraphs of the body of claim 1 as follows:

> "a first pointer, entered in the data structure, to a sequence of ~~one or more~~ commands, executable by a processor, implementing one or more packet modification operations and stored, in packed format, with more than one command stored in a single addressable entry of the stored sequence, in a first memory area; and
>
> a second pointer, also entered in the data structure, to a burst of ~~one or more~~ data or mask items, stored, in packed format, with more than one data or mask item stored in a single addressable entry of the stored burst, in a second memory area distinct from the first, for use by the processor in executing the one or more commands;"

These amendments are supported, for example, by Fig. 20, reproduced below, showing an example of the claimed data structure:



10.4.1.2. Internal Link Entry Format

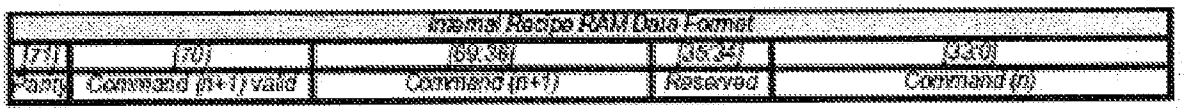
Bit	Function	Description
20-0	BURST ADDR 1	Burst Address 1.
25-21	BURST LEN 1	Burst Length 1.
46-26	BURST ADDR 2	Burst Address 2.
51-47	BURST LEN 2	Burst Length 2.
62-52	INT RECIPE INDEX	Internal Recipe Index.
67-63	INT RECIPE LEN	Internal Recipe Length.
69-68	-	Reserved.
70	INT/EXT	Internal/External.
71	PAR	Parity Bit. Set so that there is odd parity across bits 71:0 of the entry data.

FIGURE 20

As can be seen, this example of the claimed data structure includes a pointer to a sequence of commands (the Internal Recipe Index, contained in bits 52-62), and a pointer to a burst of data or mask items (either the Burst Address 1, contained in bit 0-20, or the Burst Address 2, contained in bits 26-46).

The claim has further been amended to require that the commands in the sequence pointed to by the first pointer be stored in packed format, where more than one command is stored in a single addressable entry in the stored sequence, and that the data or mask items in the burst be stored in packed format, where more than one data or mask item is stored in a single addressable entry in the burst.

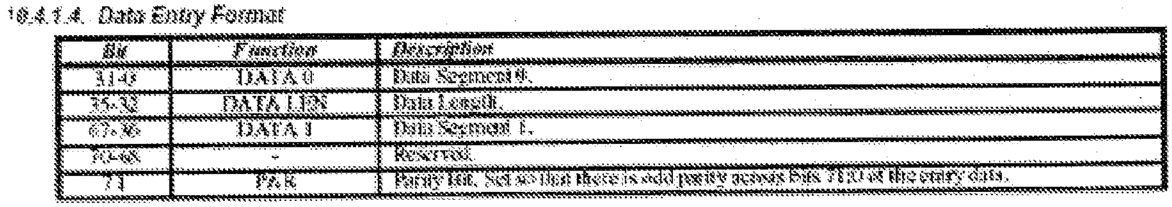
These amendments are supported, for example, by Fig. 22, reproduced below, showing more than one command in the sequence being stored in a single addressable entry of the stored sequence:



Internal Recipe RAM Data Format				
[71]	[70]	[69:36]	[35:34]	[33:0]
Parity	Command (n+1) valid	Command (n+1)	Reserved	Command (n)

FIGURE 22

It is also supported by Fig. 21B, reproduced below, showing more than one data or mask items in the burst being stored in a single addressable entry in the stored burst:



10.4.1.4. Data Entry Format

Bit	Function	Description
31-0	DATA 0	Data Segment 0.
35-32	DATA LEN	Data Length.
67-36	DATA 1	Data Segment 1.
70-68	-	Reserved.
71	PAR	Parity bit. Set so that there is odd parity across bits 71:0 of the entry data.

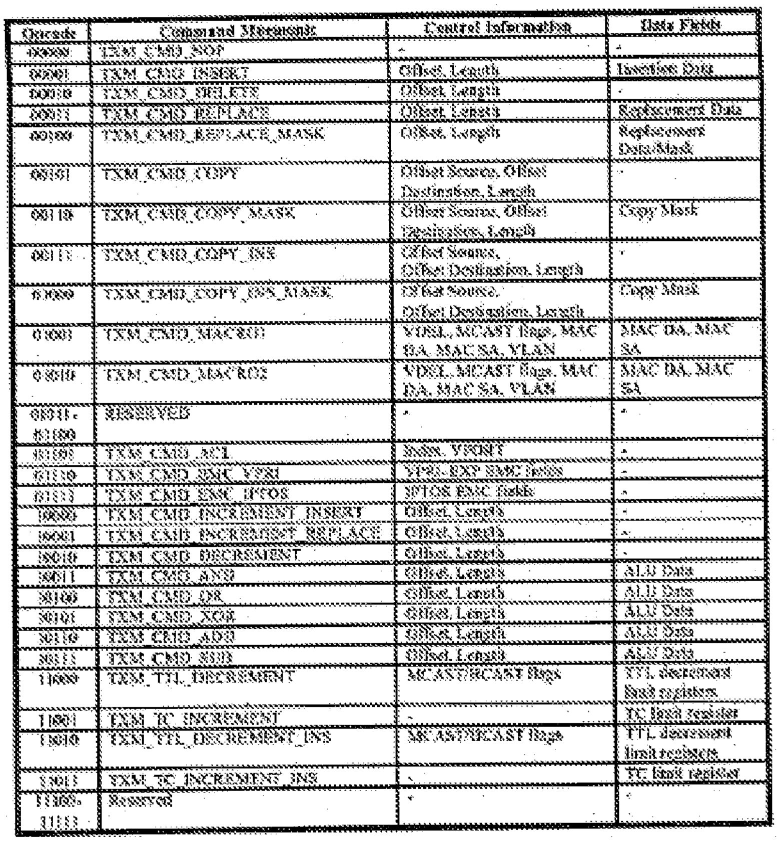
FIGURE 21A

Further support for these amendments is provided, for example, by page 31, lines 4-12, page 31, line 26, to page 32, line 2, page 31, lines 15-25.

Claim 1 has further been amended to clarify that the commands in the sequence specify performing one or more packet modification operations using, as operands or masks of operands, the data or mask items in the burst:

> "wherein ~~at least one~~the commands in the sequence ~~implements a packet modification operation that uses~~specify performing the one or more packet modification operations using, as operands or as masks for operands, ~~at least one of the one or more~~ the data or mask items in the burst~~to modify a packet~~"

This amendment is supported by Fig. 25, and related text at page 33, line 11, to page 37, line 9, illustrating and describing several examples of the commands, which utilize the data or mask items as operands or masks of operands:



Opcode	Command Mnemonic	Control Information	Data Fields
00000	TXM_CMD_NOP	-	-
00001	TXM_CMD_INSERT	Offset, Length	Insertion Data
00010	TXM_CMD_DELETE	Offset, Length	-
00011	TXM_CMD_REPLACE	Offset, Length	Replacement Data
00100	TXM_CMD_REPLACE_MASK	Offset, Length	Replacement Data/Mask
00101	TXM_CMD_COPY	Offset Source, Offset Destination, Length	-
00110	TXM_CMD_COPY_MASK	Offset Source, Offset Destination, Length	Copy Mask
00111	TXM_CMD_COPY_INS	Offset Source, Offset Destination, Length	-
01000	TXM_CMD_COPY_INS_MASK	Offset Source, Offset Destination, Length	Copy Mask
01001	TXM_CMD_MACRO1	VDEL, MCAST flags, MAC DA, MAC SA, VLAN	MAC DA, MAC SA
01010	TXM_CMD_MACRO2	VDEL, MCAST flags, MAC DA, MAC SA, VLAN	MAC DA, MAC SA
01011-01100	RESERVED	-	-
01101	TXM_CMD_ACL	Index, VRMT	-
01110	TXM_CMD_EMC_VPN	VPN EXP EMC fields	-
01111	TXM_CMD_EMC_IPTOS	IPTOS EMC fields	-
10000	TXM_CMD_INCREMENT_INSERT	Offset, Length	-
10001	TXM_CMD_INCREMENT_REPLACE	Offset, Length	-
10010	TXM_CMD_DECREMENT	Offset, Length	-
10011	TXM_CMD_AND	Offset, Length	ALU Data
10100	TXM_CMD_OR	Offset, Length	ALU Data
10101	TXM_CMD_XOR	Offset, Length	ALU Data
10110	TXM_CMD_ADD	Offset, Length	ALU Data
10111	TXM_CMD_SUB	Offset, Length	ALU Data
11000	TXM_TTL_DECREMENT	MCAST/BCAST flags	TTL decrement limit registers
11001	TXM_TC_INCREMENT	-	TC limit register
11010	TXM_TTL_DECREMENT_INS	MCAST/BCAST flags	TTL decrement limit registers
11011	TXM_TC_INCREMENT_INS	-	TC limit register
11100-11111	Reserved	-	-

FIGURE 25

Similar amendments have been made to independent claims 11 and 24.

The amendments to dependent claims 6, 9, 10, 18, 21, 22 and 30 are necessitated by the amendments to the independent claims.

As can be seen from the foregoing, none of the amendments introduce new matter.

<u>35 U.S.C. § 103(a) Rejections</u>

<u>*Independent claims 1, 11 and 24*</u>

Independent claims 1, 11 and 24 are rejected under 35 U.S.C. § 103(a) for obviousness over Okagawa, *et al.*, US Pat. Pub. No. 2007/0291754 (hereinafter "Okagawa") in view of Hung, *et al*, US Pat. No. 6,530,010 (hereinafter "Hung").

However, as amended, independent claims 1, 11 and 24 are patentable over Okagawa and Hung, considered singly or in combination. Five arguments support this.

First, Okagawa and Hung are not in analogous fields, and address different problems. Okagawa relates to the field of packet communication systems and a transfer device (*e.g.*, router) for use therein. (*See* Okagawa, par. [0001]). It concerns the lack of flexibility of conventional routers to respond to changes in the network control functions to be performed by such routers. (*Id.* at par. [0024]). In contrast to this, Hung relates to the field of signal processing, and, more specifically, to Single Instruction Multiple Data (SIMD) coprocessor architectures providing for faster image and video processing. (*See* Hung, Col. 1, lines 10-14). It concerns the inability to efficiently handle two-dimensional (2D) filtering operations using architectures configured with one-dimensional (1D) filtering operations in mind. (*Id.* at Col. 1, lines 16-40). On this basis alone, Hung is not combinable with Okagawa. (*See* MPEP §2141.01(a)(I)).

Second, there would have been no motivation for one of ordinary skill in the art, seeking to address the problem of lack of flexibility of conventional routers described in Okagawa to look to the teachings of Hung to address this problem. The Office Action claims that one of ordinary skill in the art, concerned as taught in Okagawa with sending packets of instructions to multiple routers in a packet communication system, would have been motivated to look to the teachings of Hung to represents these packets with pointers in order to reduce the number of copies of the packets that need to be transmitted, (*see* paragraph bridging pages 4-5 of the Office Action), but this is incorrect as the instructions themselves (not a pointer to these instructions) would need to be transmitted to each the routers in Ckagawa in order to change their functionality by modifying a function mapping table (shown in Fig. 5) maintained on the router, as taught by Okinawa. (*See* Okagawa, par. [0079] ("The function-mapping control unit 52 is a manager configured to update the function-mapping table 53 (first memory) <u>in accordance with the instruction information received from the network manager 30</u>.") (emphasis added)). In other words, it would not suffice to simply transmit a pointer to such instructions to the router. Therefore, there would have been no motivation to one of ordinary skill in the art to look to the teachings of Hung to represent these packets with pointers. This lack of motivation weighs against the ability to combine Okagawa with Hung to support an obviousness rejection.

Third, even assuming *arguendo* it is permissible to combine Okagawa with Hung (it isn't, but assuming for the sake of argument it is), any such combination would not meet the limitations of the claims as amended. For example, nowhere do the combined teachings of

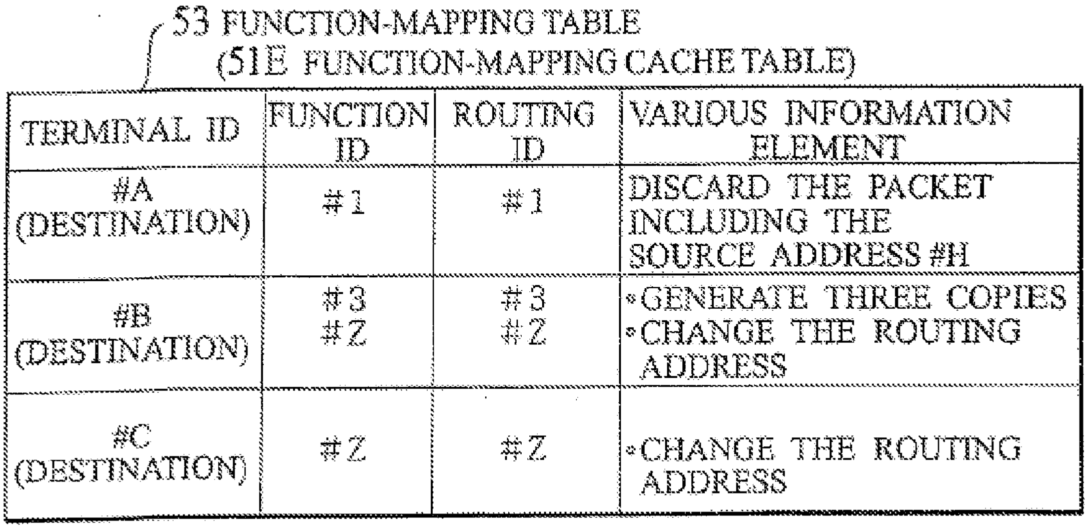
Okagawa and Hung teach or suggest the claimed data structure that comprises a first pointer, entered in the data structure, that points to a sequence of commands in a first memory area stored in packed format, with more than one command stored in a single addressable entry in the stored sequence, and a second pointer, also entered in the same data structure, that points to a burst of data or mask items in a second memory area distinct from the first stored in packed format, with more than one data or mask item stored in a single addressable entry in the stored burst, where the commands specify one or more packet modification operations using the data or mask items as operands or masks of operands in the one or more packet modification operations. These requirements are set forth in the following paragraphs of the body of claim 1 as amended, and similar amendments are recited in independent claims 11 and 24:

> "a first pointer, entered in the data structure, to a sequence of ~~one or more~~ commands, executable by a processor, implementing one or more packet modification operations and stored, in packed format, with more than one command stored in a single addressable entry of the stored sequence, in a first memory area; and
>
> a second pointer, also entered in the data structure, to a burst of ~~one or more~~ data or mask items, stored, in packed format, with more than one data or mask item stored in a single addressable entry of the stored burst, in a second memory area distinct from the first, for use by the processor in executing the one or more commands;
>
> wherein ~~at least one~~the commands in the sequence ~~implements a packet modification operation that uses~~specify performing the one or more packet modification operations using, as operands or as masks for operands, ~~at least one of the one or more~~ the data or mask items in the burst~~to modify a packet~~."

Simply representing the packet of instructions referenced in par. [0056] of Okagawa with a pointer would not achieve this data structure as the Office Action seems to assume. As mentioned, the claim requires that there be at least two pointers entered in the same data structure, the first to a sequence of commands, and the second to a burst of data or mask items, where the commands in the sequence specify one or more packet modification operations that use, as operands or mask of operands, the data or mask items in the burst. Nothing in the combined teachings of Okagawa and Hung teach or suggest a data structure with both these pointers.

Moreover, nothing in Okagawa or Hung teaches or suggests pointers that point to commands (or operands or operand masks) stored in packed format with more that one command (or operand or mask of operand) stored in a single addressable entry of a stored sequence or burst. The Office Action claims that Fig. 5 of Okagawa illustrates this feature, (*see* Office Action, page 10), but it does not. Instead, Fig. 5 of Okagawa, reproduced below, and related text (e.g., par. [0085]) merely indicate that various IDs and information elements are stored in the same table (*i.e.*, function-mapping table and/or function-mapping cache table). But nothing in this figure or related text teaches or suggests storing more than one command (or operand or operand mask) in a single addressable entry of a stored sequence or burst, which is the requirement stated in the claims as amended.

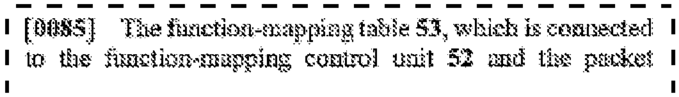
FIG. 5



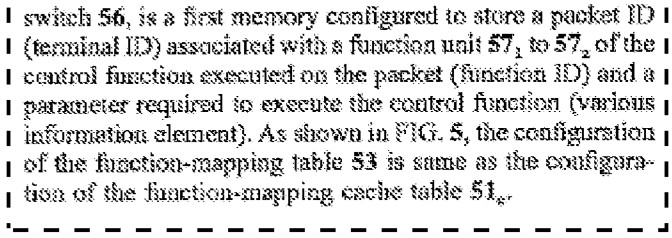
53 FUNCTION-MAPPING TABLE
(51E FUNCTION-MAPPING CACHE TABLE)

TERMINAL ID	FUNCTION ID	ROUTING ID	VARIOUS INFORMATION ELEMENT
#A (DESTINATION)	#1	#1	DISCARD THE PACKET INCLUDING THE SOURCE ADDRESS #H
#B (DESTINATION)	#3 #Z	#3 #Z	◦GENERATE THREE COPIES ◦CHANGE THE ROUTING ADDRESS
#C (DESTINATION)	#Z	#Z	◦CHANGE THE ROUTING ADDRESS

Fourth, Fig. 5 and related text (par. [0085]) of Okagawa teaches or suggests storing the various IDs and information elements in the same area of memory.



[0085] The function-mapping table 53, which is connected to the function-mapping control unit 52 and the packet



switch **56**, is a first memory configured to store a packet ID (terminal ID) associated with a function unit **57**$_1$ to **57**$_2$ of the control function executed on the packet (function ID) and a parameter required to execute the control function (various information element). As shown in FIG. **5**, the configuration of the function-mapping table **53** is same as the configuration of the function-mapping cache table **51**$_k$.

Thus, Okagawa, which is the base reference, teaches against the requirement of the claims that the sequence of commands be stored in a first memory area, and the burst of data or mask items be stored <u>in a second memory area distinct from the first</u>. On this basis as well, the obviousness rejection of Okagawa in view of Hung must fail. (*See* MPEP §2144.05(III) ("A *prima facie* case of obviousness may also be rebutted by showing that the art, in any material respect, teaches away from the claimed invention.")).

Fifth, since the principle of operation of Okagawa, which is the base reference, is to store the various IDs and information elements in the same table and, therefore, area of memory, nothing in Hung or any of the secondary references can be relied on to support deviating from this principle of operation. (*See* MPEP 2143.01(VI) ("If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious.")). For this additional reason, the obviousness rejection based on Okagawa in view of Hung must fail.

For all the foregoing reasons, independent claims 1, 11 and 24 are allowable over the cited art.

<u>Dependent claims 2-3, 6-10, 12-15, 18-22, 28-30</u>

Dependent claims 2-3, 6-10, 12-15, 18-22 and 28-30 are dependent, directly or indirectly, on independent claims 1, 11 and 24. Since claims 1, 11 and 24 are patentable over the cited art, for the reasons discussed, supra, then dependent claims 2-3, 6-10, 12-15, 18-22 and 28-30 are allowable over the cited art as well, due to their dependence on an allowable base claim.

CONCLUSION

In light of the foregoing, Applicant respectfully submits that all claims are allowable and on that basis earnestly solicits the Examiner to allow all claims and pass this application to issuance. If any issues or questions remain, the Examiner is invited to call the undersigned at (940) 759-5269.

Respectfully submitted,

Date: October 4, 2009

______/Robert C. Laurenson/________
Robert C. Laurenson (Reg. No. 34,206)

HOWREY LLP
2941 Fairview Park Drive, Box 7
Falls Church, VA 22042
Tel: 949.721.5269
Fax: 949.721.6910